REMARKS

Claims 1-11, 13-20, 23, and 37-48 are pending in the application. Claims 1-11, 13-20, 23, and 37-48 stand rejected. Applicants herein amend claims 1, 37, and 43 to clarify the claimed subject matter. Applicants request further review and examination in view of the amendments and the following remarks.

Summary of Examiner Interview

On November 18, 2010 the undersigned attorney of record conducted an interview. During the interview the 35 U.S.C. § 112 and 35 U.S.C. § 103 rejections were discussed. The Examiner agreed that the amendments to claims 1, 37, and 43 place the claims in condition for allowance and that a Notice of Allowance of all pending claims would be forthcoming after the submission of this response.

Specification

Applicants understand that the specification is lengthy and will cooperate with the Examiner to help identify any possible spelling and grammatical errors.

Claim Rejections – 35 USC § 112

Claims 1, 37, and 43 stand rejected under 35 U.S.C. § 112, first paragraph, as allegedly failing to comply with the written description requirement. Applicants traverse these rejections.

Regarding claim 1, the Office asserts the following elements of claim 1 are not supported by the written description:

> the instructions for the database management program further including instructions for receiving receive read/write requests from the user mode applications for Items via one or more functions of an operating system application program interface; and
>
> the instructions for the database management program further including instructions for deserializing files storing the file data for

> the Items into Items and return the Items to the user mode applications

The Office asserts that the first element is not supported by the written description stating that "there is no support for the claimed limitations…it is not clear where this is even disclosed in the cited paragraphs." (Office Action at p. 3). A claim fails to comply with the written description requirement when a patent specification does not describe the claimed invention in sufficient detail that one skilled in the art can reasonably conclude that the inventor had possession of the claimed invention. See, e.g., *Moba, B.V. v. Diamond Automation, Inc.*, 325 F.3d 1306, 1319, 66 USPQ2d 1429, 1438 (Fed. Cir. 2003); *Vas-Cath, Inc. v. Mahurkar*, 935 F.2d at 1563, 19 USPQ2d at 1116. The Office has the initial burden of presenting evidence or reasoning, under a preponderance of the evidence standard, that a person of skill in the art would not recognize that the written description of the invention provides support for the claims. Literal support for a limitation is not required; rather, the MPEP recognizes that "newly added claim limitations must be supported in the specification through *express, implicit, or inherent* disclosure." (MPEP section 2163) (Emphasis added).

When viewing the rejection of claim 1 with this standard in mind, it is clear that the Office has not proved that a person of skill in the art that the time of invention would not recognize that the written description of the invention provides support for the claims. The assertion above is nothing more than a conclusory statement; far from the required showing of *evidence or reasoning* that shows why the limitation is not supported. This lack of analysis is insufficient to prove that the subject matter at issue is "new matter." Rather, the Office must put forth evidence or reasoning that proves that the claimed subject matter is not supported by express, implicit, or inherent disclosure. Consequently, the Office has not properly demonstrated, *by evidence*, that the claimed subject matter is not supported by the specification. Accordingly, Applicants request that this rejection of claims 1 be withdrawn for this reason.

Moreover, Applicants submit that the specification supports the claimed subject matter. First, Applicants' note that the specification describes the terms "Item" and "object" as follows:

> [0102] An *Item* is a unit of storable information that, unlike a simple file, *is an object* having a basic set of properties that are commonly supported across all objects exposed to an end-user or

> application program by the storage platform. Items also have properties and relationships that are commonly supported across all Item types including features that allow new properties and relationships to be introduced, as discussed below. (Emphasis added).
>
> [0109] *Items are stand-alone objects*; thus, if you delete an Item, all of the Items immediate and inherited properties are also deleted. Similarly, when retrieving an Item, what is received is the Item and all of its immediate and inherited properties (including the information pertaining to its complex property types). Certain embodiments of the present invention may enable one to request a subset of properties when retrieving a specific Item; however, the default for many such embodiments is to provide the Item with all of its immediate and inherited properties when retrieved. Moreover, the properties of Items can also be extended by adding new properties to the existing properties of that Item's type. These "extensions" are thereafter bona fide properties of the Item and subtypes of that Item type may automatically include the extension properties. (Emphasis added).
>
> [0723] *The storage platform API exposes common actions on all items*--Create, Delete, Update; *these are exposed as methods on objects*. In addition, domain specific actions such as SendMail, CheckFreeBusy, etc. are also available as methods. The API framework uses well defined patterns that ISVs can use to add value by defining additional actions. (Emphasis added).
>
> [0870] Once an object has been retrieved by a search it may be modified by the application as needed. New *objects* may also be created and associated with existing objects. Once the application has made all the changes that form a logical group, the application calls ItemContext.Update to persist those changes to the store. According to yet another aspect of the storage platform API of the present invention, *the API collects changes to an item made by an application program and then organizes them into the correct updates required by the database engine* (or any kind of storage engine) on which the data store is implemented. This enables application programmers to make changes to an item in memory, while leaving the complexity of data store updates to the API. (Emphasis added).

The specification seemingly uses the terms Item and object to describe the same data. However, when the data is described as the subject of an operation it is described using object oriented techniques, e.g., as methods on objects.

The specification describes an application program interface throughout the specification as an interface and in particular at paragraphs [0088] and [0714], which state:

> [0088] The storage platform of the present invention still further comprises an application programming interfaces (API) 322, which *enables application programs, such as application programs 350a, 350b, and 350c, to access all of the foregoing capabilities of the storage platform* and to access the *data described in the schemas*. The storage platform API 322 may be used by application programs in combination with other APIs, such as the OLE DB API 324 and the Microsoft Windows Win32 API 326. (Emphasis added)
>
> [0714] As mentioned above, the storage platform comprises an API that enables application programs to access the features and capabilities of the storage platform discussed above and *to access items stored in the data store*. This section describes one embodiment of a storage platform API of the storage platform of the present invention. (Emphasis added).

Clearly, the text of these two paragraphs show that an application program interface enables applications to access the items stored in the data store. The application describes the interaction between the API and the database in more detail in subsequent paragraphs as using a database management program:

> [0715] FIG. 19 illustrates the basic architecture of the storage platform API, in accordance with the present embodiment. T*he storage platform API uses SQLClient 1900 to talk to the local data store 302* and may also use SQLClient 1900 to talk to remote data stores (e.g., data store 340). The local store 302 may also talk to the remote data store 340 using either DQP (Distributed Query Processor) or through the the storage platform synchronization service ("Sync") described above. The storage platform API 322 also acts as the bridge API for data store notifications, passing application's subscriptions to the notification engine 332 and routing notifications to the application (e.g., application 350a, 350b, or 350c), as also described above. In one embodiment, the storage platform API 322 may also define a limited "provider"

> architecture so that it can access data in Microsoft Exchange and AD.

Based on the aforementioned paragraphs, Applicants respectfully submit that the specification includes support for a database management program that receives requests for items from an application program interface.

Turning to paragraph [0778], it describes exemplary techniques that can be used by the API to access the database. In particular, this paragraph supports the notion that the request can be to "read/write." Specifically, this paragraph states:

> [0778] The basic storage platform API programming model is object persistence. Application programs (or "applications") execute a search on a store and *retrieve objects representing the data in the store*. Applications *modify* the retrieved objects or *create* new objects, then cause their changes to be propagated into the store. This process is managed by an ItemContext object. Searches are executed using an ItemSearcher object and search results are accessible via a FindResult object. (Emphasis added).

This paragraph specifically identifies the storage platform API can use a function ItemContext to access the database to "retrieve objects," "modify the retrieved objects," and "create new objects." Applicants submit that the Office has not demonstrated that a person of skill in the art would fail to appreciate that a disclosure of an API that receives requests from applications and issues commands to a database management program to retrieve, modify, and create objects supports "receiving read/write requests" as claimed.

The analysis presented above has shown that the specification supports the subject matter: a database management program that receives requests from an API, that the requests from the API can be to read/write to objects, and that the API exposes common actions on items as methods on objects.

Applicants respectfully submit that the following paragraphs provide support for the element:

> the instructions for the database management program further including instructions for deserializing files storing the file data for the Items into Items and return the Items to the user mode applications

ItemContext, which is described as including a runtime framework that includes ItemContext, is described in the following terms:

> [0782] An ItemContext object (i) represents a set of item domains that an application program wants to search, (ii) maintains state information for each object that represents the state of the data as retrieved from the storage platform, and (iii) *manages the transactions used when interacting with the storage platform and any file system with which the storage platform may interoperate.*

In subsequent paragraphs, ItemContext is described as capable of performing the following services:

> [0784] 1. *Deserializes data read from the store into objects.*
>
> [0790] 3. Insures that *file backed items are properly updated* when changes to the object(s) representing that item are saved.
>
> [0792] 5. Performs *item creation, copy, move, and delete operations that take storage platform relationship semantics, file backed items*, and stream typed properties into account.

Applicants respectfully submit that these paragraphs describe the subject matter as claimed. Specifically, these paragraphs describe deserializing data, e.g., file backed items, from the store into objects. The analysis above has shown that common actions on items are described as methods on objects. Consequently, the deserializing operations performed by ItemContext are to extract data in files and generate items. In contrast to Applicants' analysis, the Office has failed to articulate a reason, backed by evidence, that proves that these paragraphs fail to support the claimed subject matter. Consequently, the Office has failed to demonstrate that these claims are not supported by the written description requirement. Accordingly, Applicants respectfully request reconsideration of the 35 U.S.C. § 112, first paragraph rejection of claim 1.

Claims 37 and 43 stand rejected under 35 U.S.C. § 112, first paragraph for the same reasons as claim 1. Claims 37 and 43 include subject matter that is similar to the subject matter recited in claim 1. Accordingly, Applicants respectfully request reconsideration of the 35 U.S.C. § 112 for the same reasons set forth above with respect to claim 1, *mutatis mutandis.*

Claim Rejections – 35 USC § 101

The Office lists claims 1-11, 13-20, 23, and 43-48 under a section entitled "Claim Rejections – 35 USC § 101," however, as far as Applicants can discern, these claims are not rejected under 35 USC § 101. (See Office Action at p. 4). Applicants agree with the Office that these claims fall within a statutory class of invention and respectfully request that the Office clarify the record by removing any indication that these claims are rejected under 35 USC § 101 by removing the claims from the section entitled "Claim Rejections – 35 USC § 101."

Claim Rejections – 35 USC § 103

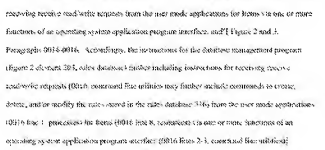
Claims 1, 37, and 43 stand rejected under 35 U.S.C. § 103(a) over U.S. Patent No. 6,018,342 to Bristor in view of U.S. Patent Application Publication No. 2003/0009685 to Choo and U.S. Patent No. 7,158,962 to Nelson. Applicants respectfully traverse these rejections.

Applicants respectfully submit that the art of record fails to teach or suggest at least:

> the instructions for the database management program further including instructions for receiving *at least Item creation, copy, move, and delete operation* from the user mode applications for Items via one or more functions of an operating system application program interface; and
>
> the instructions for the database management program further including instructions for *deserializing files storing the file data for the Items into Items and returning the Items to the user mode applications*;

Support for this amendment is found in the paragraphs described above.

Regarding this portion of claim 1, the Office asserts that Choo teaches:



"the instructions for the database management program further including instructions for receiving receive read/write requests from the user mode applications for Items via one or more functions of an operating system application program interface; and"[Figure 2 and 3. Paragraphs 0014-0016. Accordingly, the instructions for the database management program (figure 2 element 205, rules database) further including instructions for receiving receive read/write requests (0016, command line utilities may further include commands to create, delete, and/or modify the rules stored in the rules database 335) from the user mode applications (0016 line 1, processes) for Items (0016 line 8, resources) via one or more functions of an operating system application program interface (0016 lines 2-3, command line utilities)]

(Office Action at p. 8).

Choo also fails to teach or suggest the claimed subject matter. Choo describes a database with access rules. A kernel module can use the access rules to determine whether a user space application is authorized to access a file. Applicants submit that this does not teach or suggest the claimed subject matter and the Office agreed during the Interview.

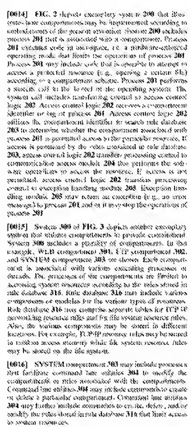
The Office asserts that figures 2-3 describes the aforementioned feature. Applicants disagree. FIG. 2 shows a process flow illustrating a method of operation. A portion of the text associated with FIG. 2 states:

> Process 201 performs a system call to the kernel of the operating system. The system call includes transferring control to access control logic 202. Access control logic 202 receives a compartment identifier or tag of process 201. Access control logic 202 utilizes the compartment identifier to search rule database 203 to determine whether the compartment associated with process 201 is permitted access to the particular resource. If access is permitted by the rules contained in rule database 203, access control logic 202 transfers processing control to communication access module 204 that performs the software operations to access the resource. If access is not permitted, access control logic 202 transfers processing control to exception handling module 205. Exception handling module 205 may return an exception (e.g., an error message) to process 201 and/or it may stop the operations of process 201. (Paragraph [0014]).

Applicant submits that nothing in the above describes the aforementioned text of claim 1. Choo describes a security module 204 that performs the software operations to access the resource. The security module 204 merely looks for rules in the database and uses them to enforce access restrictions. This fails to describe the subject matter of claim 1. In particular, the rules database allows access to a file; however, nothing in this section describes *deserializing* file data into Items, e.g., objects, and returning them to user space applications.

Similarly, FIG. 3 also fails to illustrate this feature of claim 1. FIG. 3 shows the automatic linking system. The figure shows that the rule database 316 is in the kernel; however, it does not state that file data is deserialized into Items and returned to user space applications.

Furthermore, paragraphs [0014]-[0016] fail to describe the aforementioned text of claim 1. The text of these paragraphs describe something quite different than the text of claim 1. Specifically, these paragraphs describe FIG. 2 and FIG. 3 and state:



[0014] **FIG. 2** depicts exemplary system **200** that illustrates how compartments may be implemented according to embodiments of the present invention. System **200** includes process **201** that is associated with a compartment. Process **201** executes code in user-space, i.e. a hardware-enforced operating mode that limits the operations of process **201**. Process **201** may include code that is operable to attempt to access a protected resource (e.g., opening a certain file) according to a compartment scheme. Process **201** performs a system call to the kernel of the operating system. The system call includes transferring control to access control logic **202**. Access control logic **202** receives a compartment identifier or tag of process **201**. Access control logic **202** utilizes the compartment identifier to search rule database **203** to determine whether the compartment associated with process **201** is permitted access to the particular resource. If access is permitted by the rules contained in rule database **203**, access control logic **202** transfers processing control to communication access module **204** that performs the software operations to access the resource. If access is not permitted, access control logic **202** transfers processing control to exception handling module **205**. Exception handling module **205** may return an exception (e.g., an error message) to process **201** and/or it may stop the operations of process **201**.

[0015] System **300** of **FIG. 3** depicts another exemplary system that utilizes compartments to provide containment. System **300** includes a plurality of compartments. In this example, WEB compartment **301**, FTP compartment **302**, and SYSTEM compartment **303** are shown. Each compartment is associated with various executing processes or threads. The processes of the compartments are limited to accessing system resources according to the rules stored in rule database **316**. Rule database **316** may include various components or modules for the various types of resources. Rule database **316** may comprise separate tables for TCP/IP networking resource rules and for file system resource rules. Also, the various components may be stored in different locations. For example, TCP/IP resource rules may be stored in random access memory while file system resource rules may be stored on the file system.

[0016] SYSTEM compartment **303** may include processes that facilitate command line utilities **304** to modify the compartments or rules associated with the compartments. Command line utilities **304** may include commands to create or delete a particular compartment. Command line utilities **304** may further include commands to create, delete, and/or modify the rules stored in rule database **316** that limit access to system resources.

After carefully reviewing this text, Applicants submit that it fails to describe the aforementioned claim subject matter. Moreover, the record lacks and evidence linking this text to the claimed subject matter nor is any evidence provided that proves that one of skill in the art *could and would* modify the subject matter described in the text to arrive at the claimed subject matter. Consequently, the cited sections of Bristor, Choo, and Nelson do not render claim 1 *prima facie* obvious. Accordingly, Applicants respectfully request reconsideration of the rejection of claim 1.

Insomuch as independent claims 37 and 43 recite subject matter similar to these claims, these claims define over the cited art of record for at least similar reasons as claim 1.

Accordingly, Applicants respectfully request reconsideration of the rejections of claims 37 and 43 for the reasons stated above with respect to claim 1.

Claims 2-3, 5, 7-8, 10, 38-39, 41, 44, 45, and 47 stand rejected under 35 USC § 103 as allegedly being unpatentable over Bristor in view of Choo, Nelson, and further view of US Pat. App. Pub. 2004/0199521 ("Anglin"). Applicants respectfully submit that the rejected claims depend from independent claims 1, 37, or 43. The cited portions of Anglin are not relied upon to cure the deficiencies of Bristor, Nelson and Choo noted above and Applicants submit that they do not. Accordingly, Applicants respectfully request withdrawal of these rejections under 35 USC § 103.

Claims 4, 6, 9 and 11 stand rejected under 35 USC § 103 as allegedly being unpatentable over Bristor in view of Choo, Nelson, and in further view of US Pat. App. Pub. 2004/0073560 ("Edwards"). Applicants respectfully submit that the rejected claims depend from independent claim 1. The cited portions of Edwards are not relied upon to cure the deficiencies of Bristor, Nelson and Choo noted above and Applicants submit that they do not. Accordingly, Applicants respectfully request withdrawal of the rejections under 35 USC § 103.

Claims 13-20 stand rejected under 35 USC § 103 as allegedly being unpatentable over Bristor in view of Choo, Nelson and in further view of US Pat. No. 6,578,046 ("Chang"). Applicants respectfully submit that the rejected claims depend from independent claims 1. The cited portions of Chang are not relied upon to cure the deficiencies of Bristor, Nelson and Choo noted above and Applicants submit that they do not. Accordingly, Applicants respectfully request withdrawal of the rejections under 35 USC § 103.

Claim 23 stands rejected under 35 USC § 103 as allegedly being unpatentable over Bristor in view of Choo, Nelson and in further view of US Pat. No. 6,438,545 ("Beauregard"). Applicants respectfully submit that the rejected claims depend from independent claims 1. The cited portions of Beauregard are not relied upon to cure the deficiencies of Bristor, Nelson and Choo noted above and Applicants submit that they do not. Accordingly, Applicants respectfully request withdrawal of the rejections under 35 USC § 103.

Claims 40, 42, 46, and 48 stand rejected under 35 USC § 103 as allegedly being unpatentable over Bristor in view of Choo, Nelson, and in view of Anglin and in further view of Edwards. Applicants respectfully submit that the rejected claims depend from independent claims 1. The cited portions of Anglin and Edwards were not relied upon to cure the deficiencies of Bristor, Nelson and Choo noted above and Applicants submit that they do not. Accordingly, Applicants respectfully request withdrawal of the rejections under 35 USC § 103.

CONCLUSION

Applicants request the Examiner reconsider the rejections and issue a Notice of Allowance of all the claims.

Date: December 17, 2010

/David M. Platz/
David M. Platz
Registration No. 60,013

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439